



Docker & Microservice

谭宗威@格尔智慧

关于我们

成立于2014.5.12

智慧护理提供商

四条产品线

团队规模60+

行业介绍

提灯女神-弗洛伦斯·南丁格尔

以生物-心理-社会医学模式为指导思想的
单独学科

护理工作大约占了医疗工作的70%

在医疗体系中地位相对较低

从业人员380万，女性占98%

我们想做点什么？

护理发展的推动者

全民健康的守护者

以互联网+ 的方式来实现上述目标

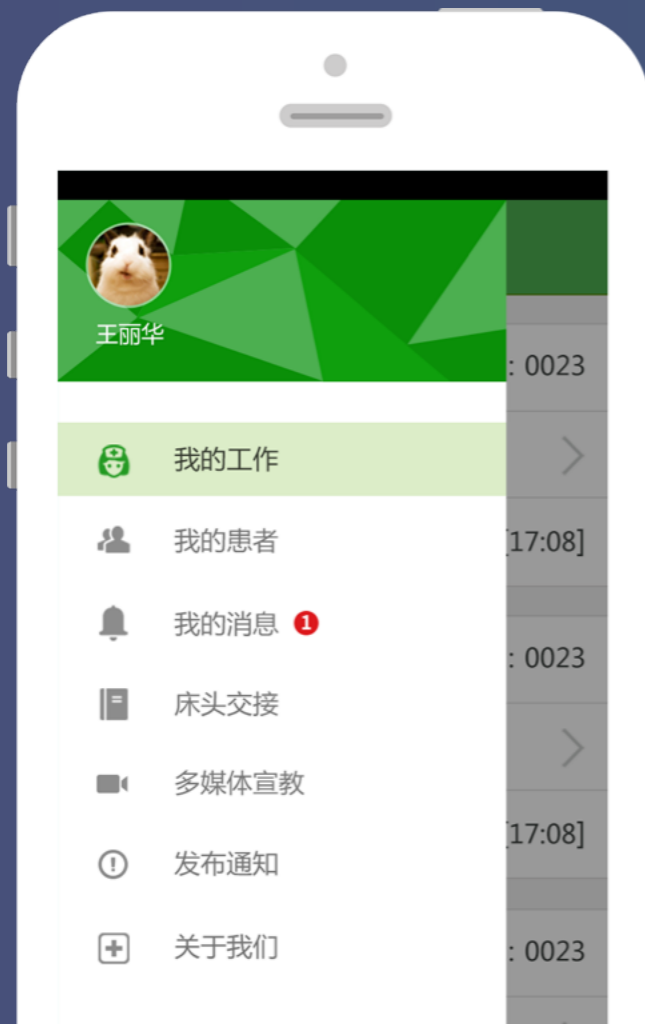
产品线

临床护理

延续护理

护士社区

知识图谱



面对的困难

业务流程复杂

业务不能停机

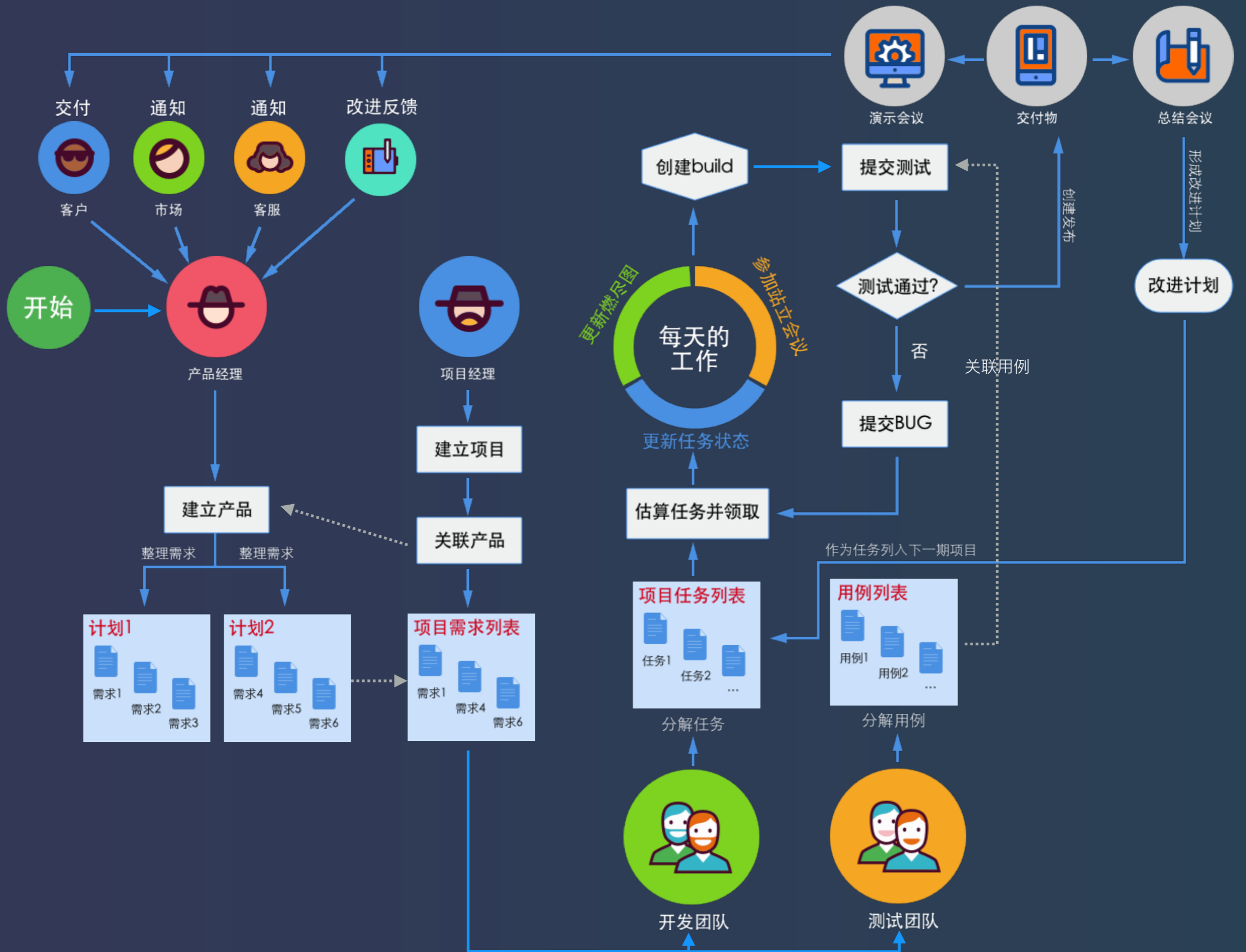
个性化要求多

部署环境复杂

流程

Scrum

迭代周期两周



工具



EA



禅道



Coding.net



阿里云



青云



DaoCloud.io

方法

用例驱动

领域驱动

面向对象

模型介绍

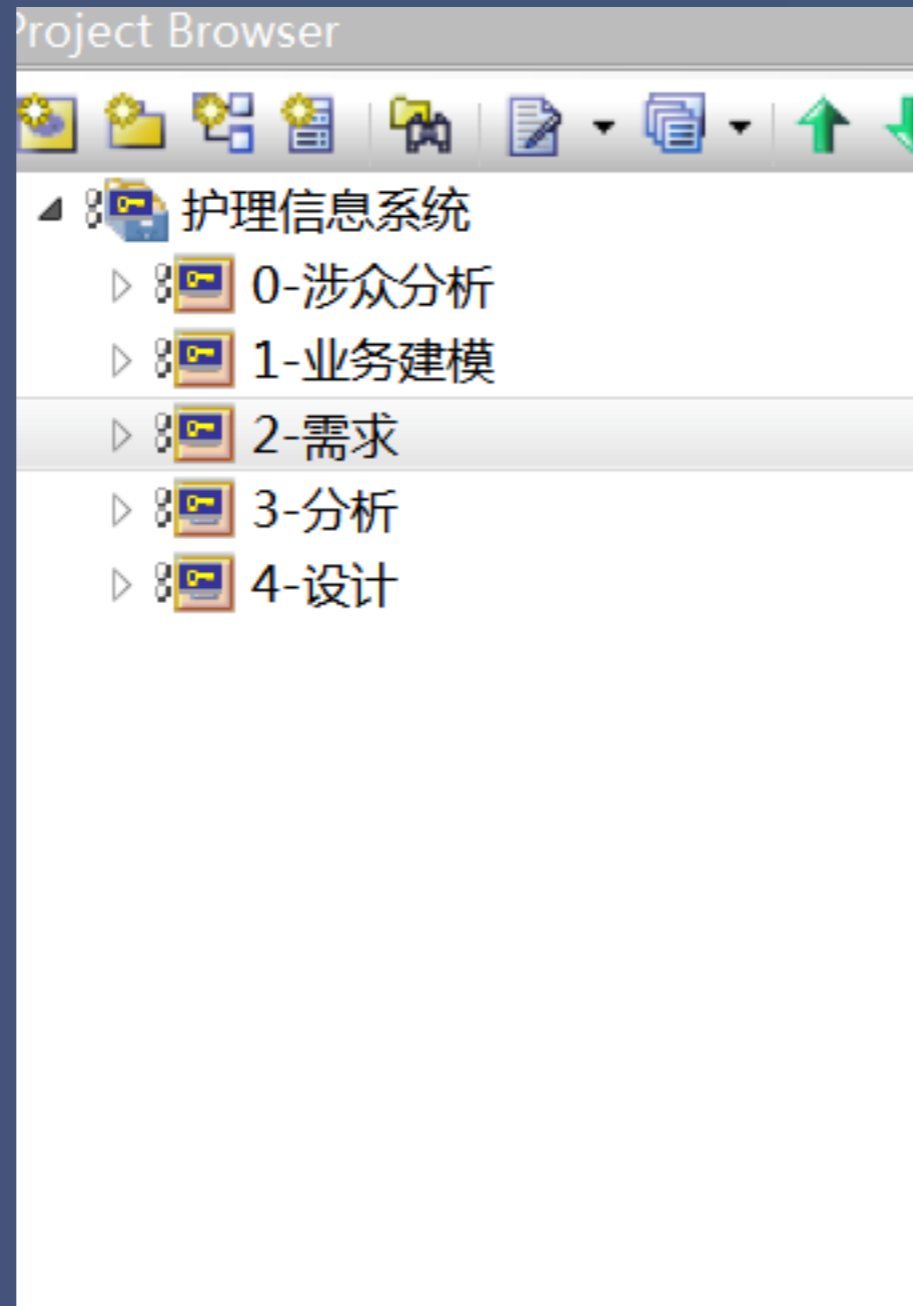
业务模型

系统模型

分析模型

领域模型

业务模型

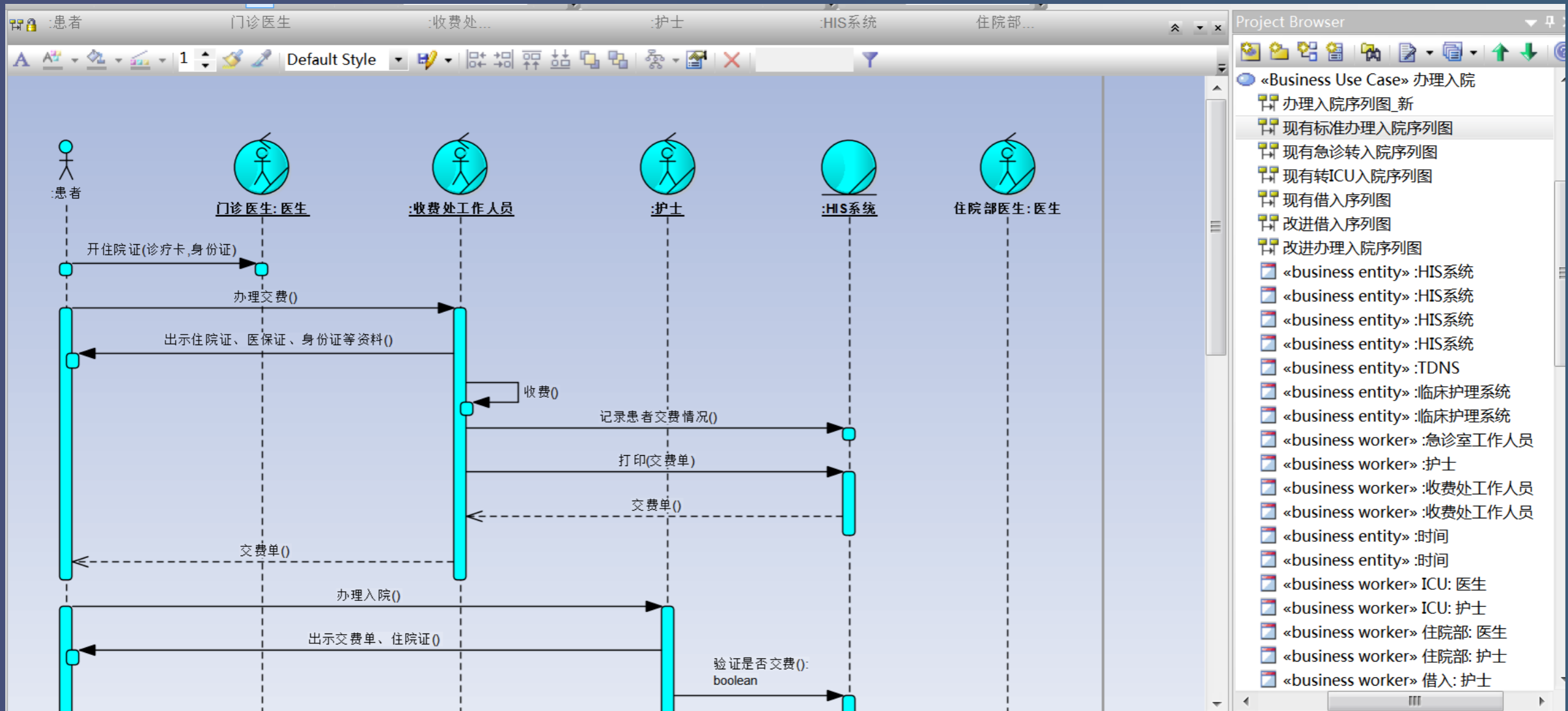


业务模型

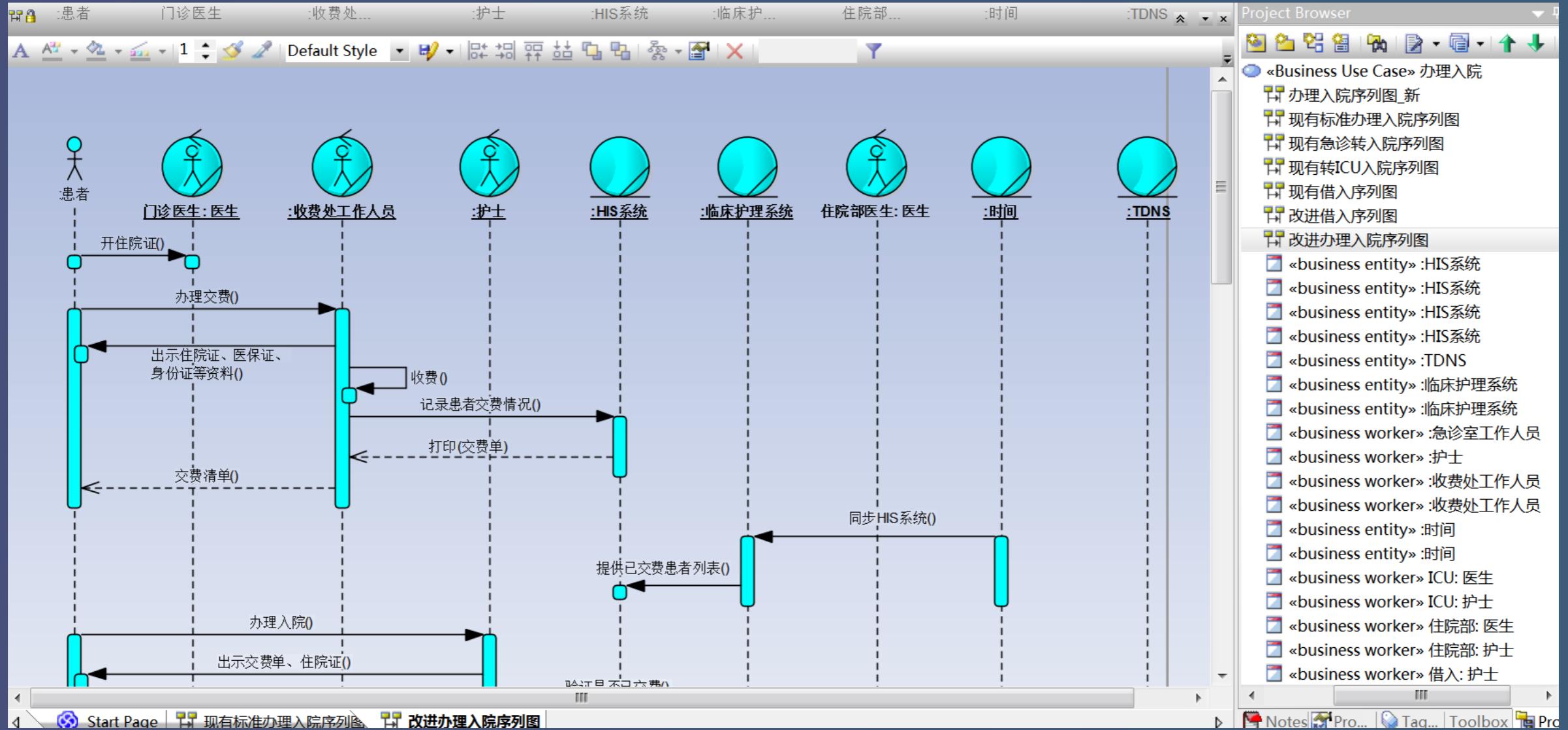
The screenshot shows a 'Project Browser' window with a toolbar at the top containing icons for home, folders, search, and navigation. The main area displays a tree view of a project structure:

- 护理信息系统
 - 0-涉众分析
 - 1-业务建模
 - 患者服务端
 - 护士社交系统
 - 护理科研与教学
 - 临床护理系统
 - 业务主角
 - 业务对象
 - 业务用例
 - 业务用例
 - «Business Use Case» 开医嘱
 - «Business Use Case» 办理入院
 - «Business Use Case» 办理出院
 - «Business Use Case» 药物医嘱治疗
 - «Business Use Case» 检查医嘱治疗
 - «Business Use Case» 检验医嘱治疗
 - «Business Use Case» 输血医嘱治疗
 - «Business Use Case» 手术医嘱治疗
 - «Business Use Case» 治疗医嘱治疗
 - «Business Use Case» 制定护理计划
 - «Business Use Case» 管道护理
 - «Business Use Case» 健康宣教

业务模型



业务模型



业务模型

UseCase : 办理入院

Properties

- General
- Tagged Values

Rules

- Requirements
- Constraints
- Scenarios

Related

- Files
- Links

Name: 办理入院

Stereotype: Business Use Case

Status: Proposed

Alias:

Complexity: Easy

Keywords:

Version: 1.0

Author: usr1

Phase: 1.0

Language: <none>

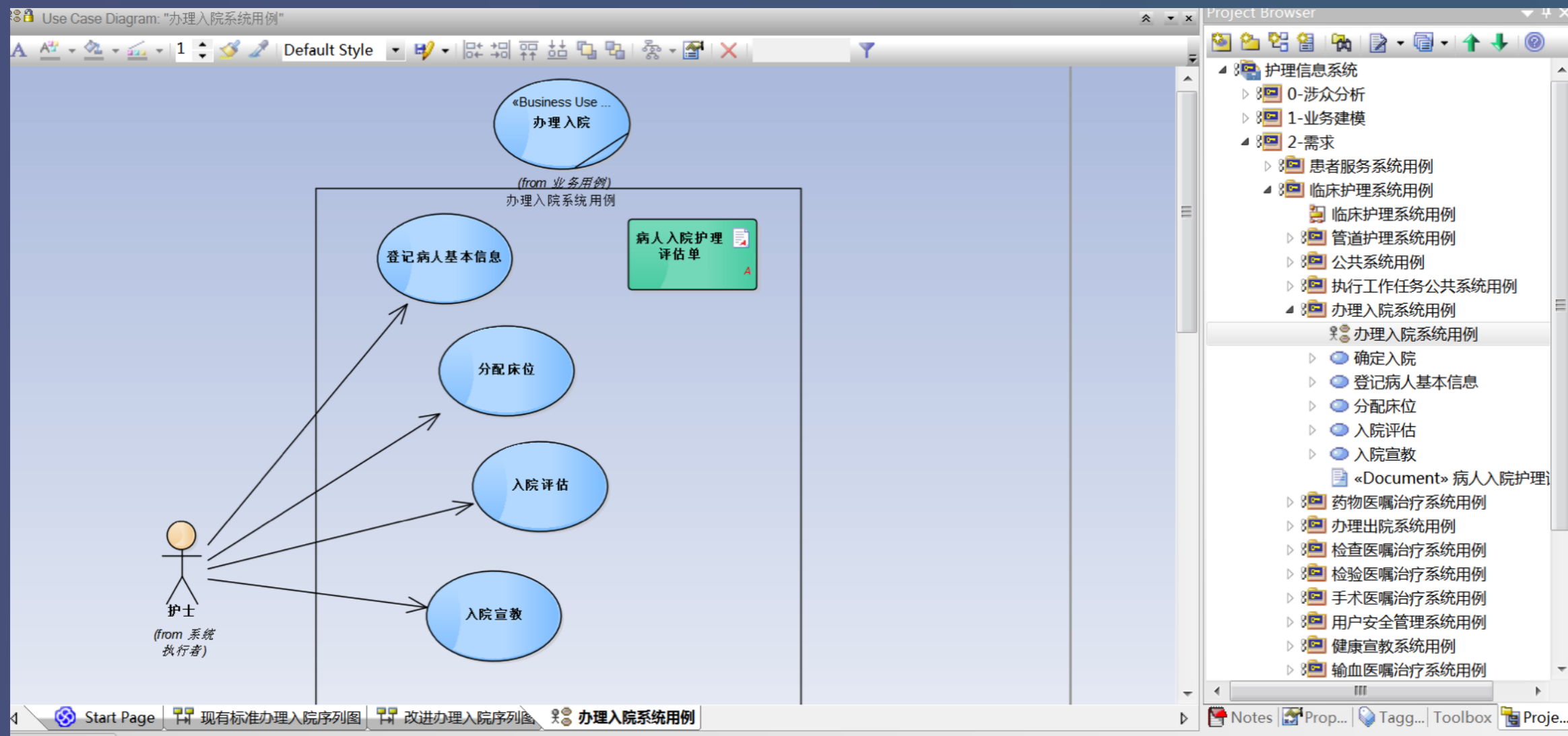
Notes:

B I U A | | | | | |

为患者办理入院手续并分配床位，进行入院宣教、评估等
涉众利益：
患者：
1. 担心入院手续太麻烦，耽误太长时间；
2. 担心不清楚医院的相关规定；
3. 担心分配的房间不合意（异性、太吵等等）；
护士：
1. 担心入院手续办理时间过长，使患者不满意；
2. 担心遗漏入院手续相关事宜（宣教、评估）；
3. 担心床位安排出错（床位被同时安排多人）；

确定 取消 应用(A) 帮助

需求模型



需求模型

The image displays a software development environment with two main windows: a UseCase editor and a Project Browser.

UseCase: 入院评估

Scenario: 基本路径 Type: Basic Path

Description: Structured Specification

S..	Action	Uses	Results	State
1	点击入院评估			
2	显示入院评估页面			
3	录入生理参数			
4	检验是否合理			
5	提交入院评估			

Entry Points | Context References | Constraints

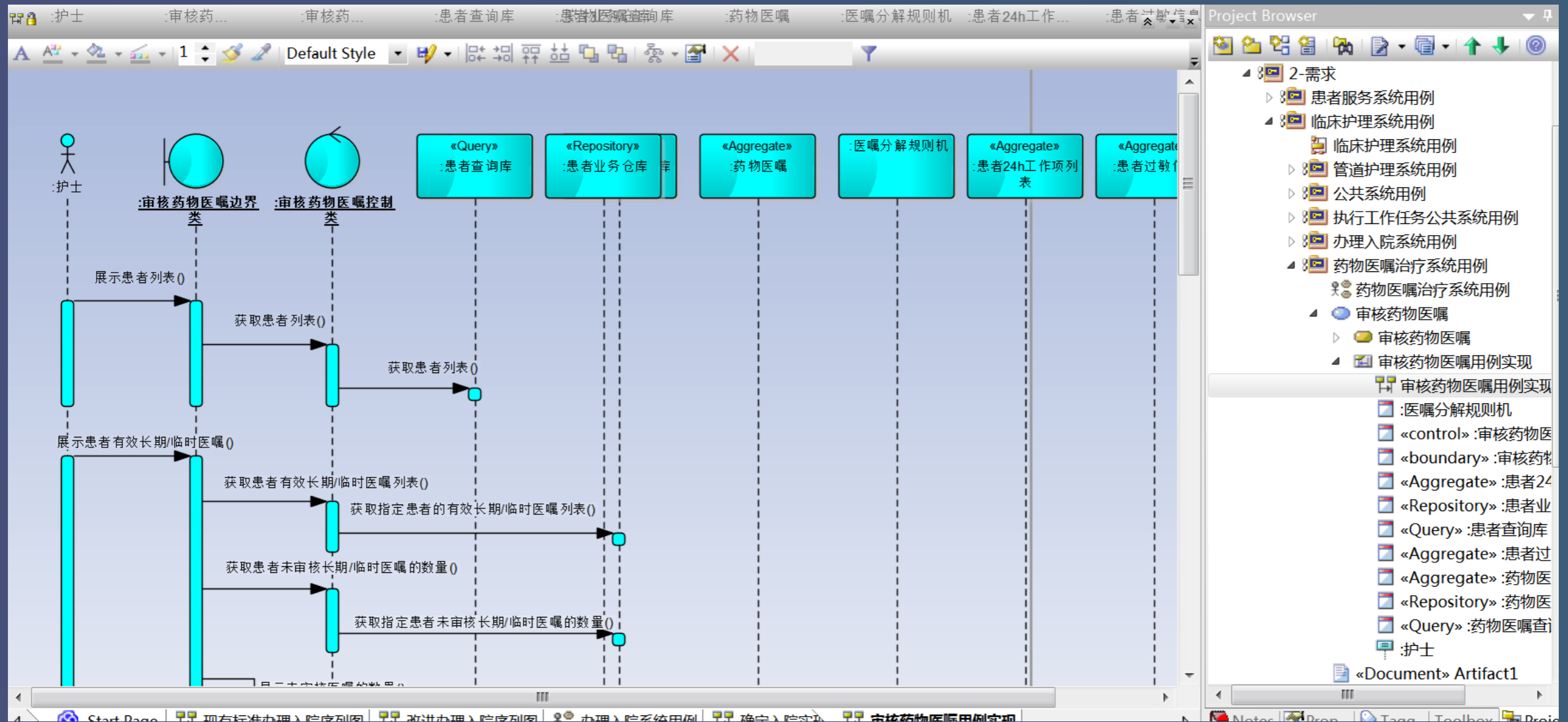
Step	Path Name	Type	Join
0	基本路径	Basic Path	-
4a	生理参数不合理	Exception	2

Buttons: 确定, 取消, 应用(A), 帮助

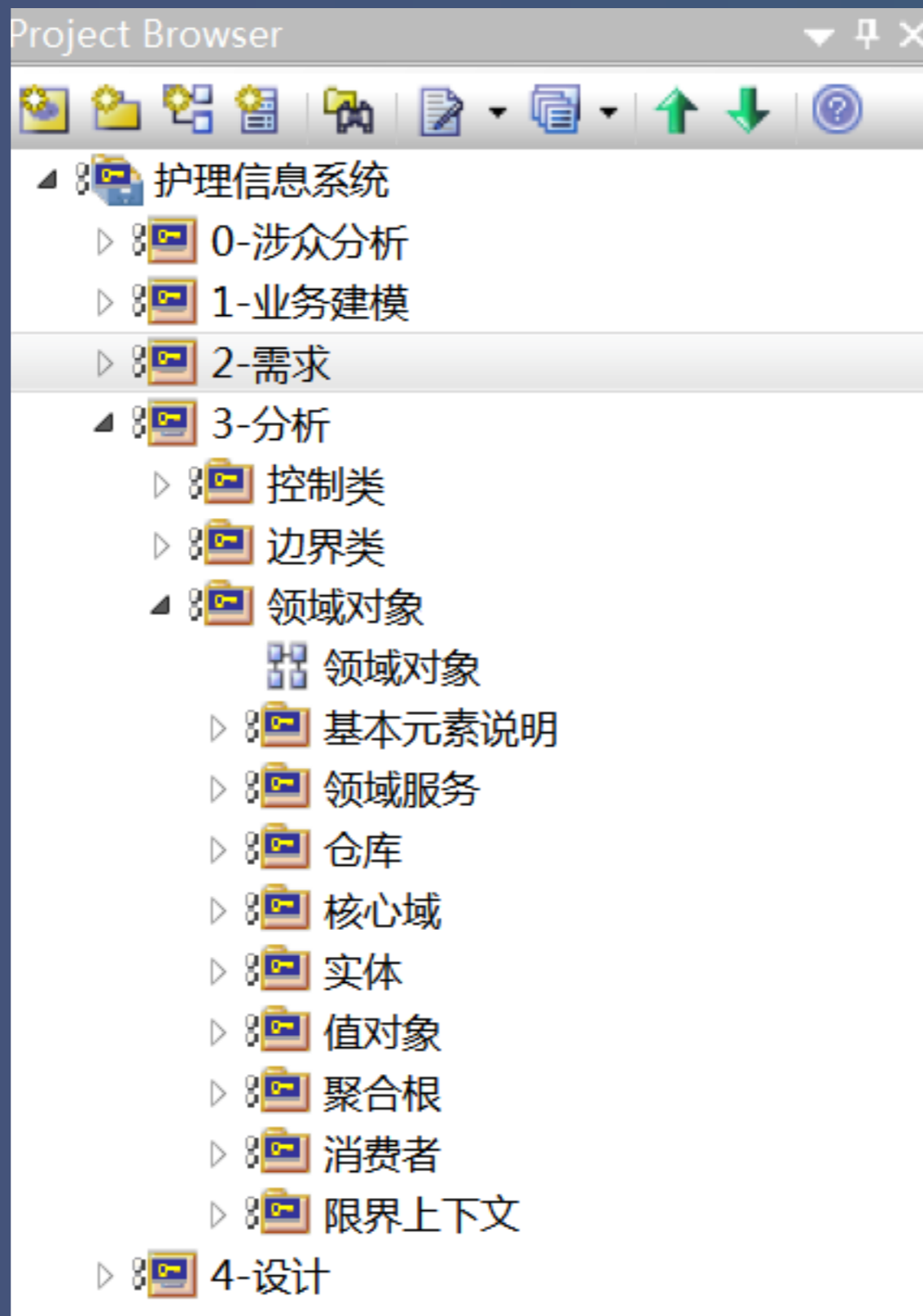
Project Browser

- 护理信息系统
 - 0-涉众分析
 - 1-业务建模
 - 2-需求
 - 患者服务系统用例
 - 临床护理系统用例
 - 临床护理系统用例
 - 管道护理系统用例
 - 公共系统用例
 - 执行工作任务公共系统用例
 - 办理入院系统用例
 - 办理入院系统用例
 - 确定入院
 - 登记病人基本信息
 - 分配床位
 - 入院评估
 - 入院宣教
 - «Document» 病人入院护理
 - 药物医嘱治疗系统用例
 - 办理出院系统用例
 - 检查医嘱治疗系统用例
 - 检验医嘱治疗系统用例
 - 手术医嘱治疗系统用例
 - 用户安全管理系统用例
 - 健康宣教系统用例

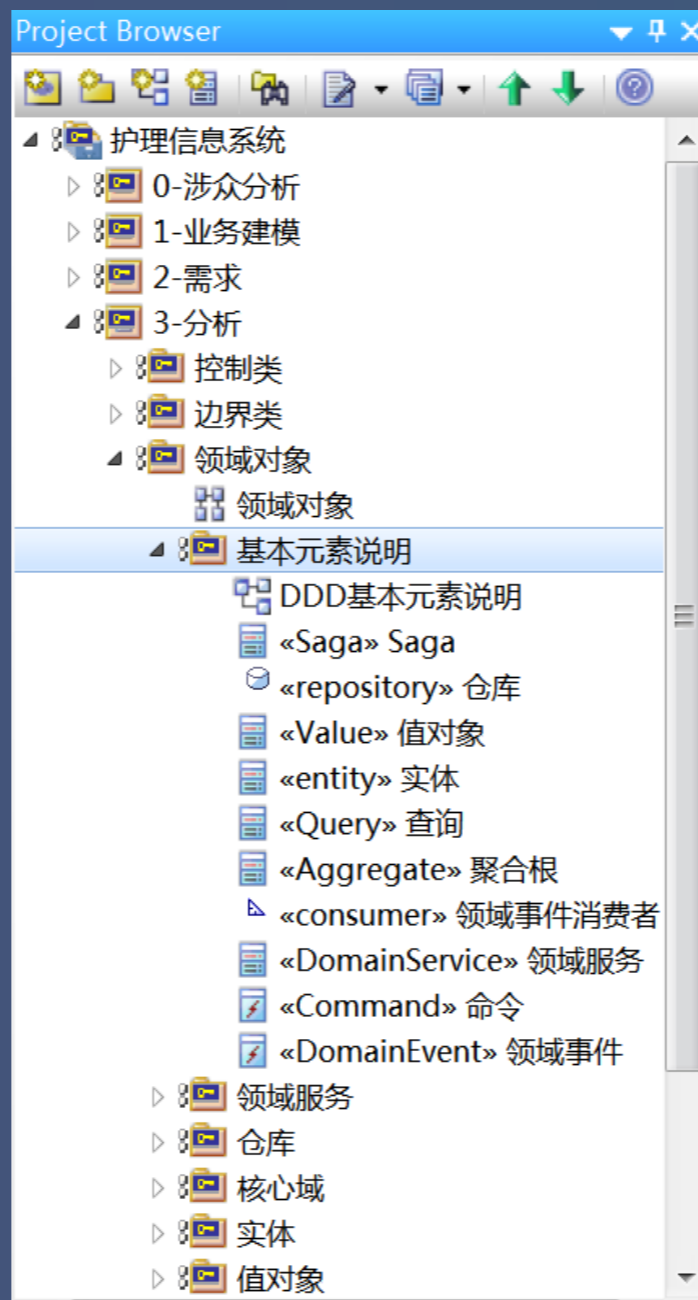
需求模型



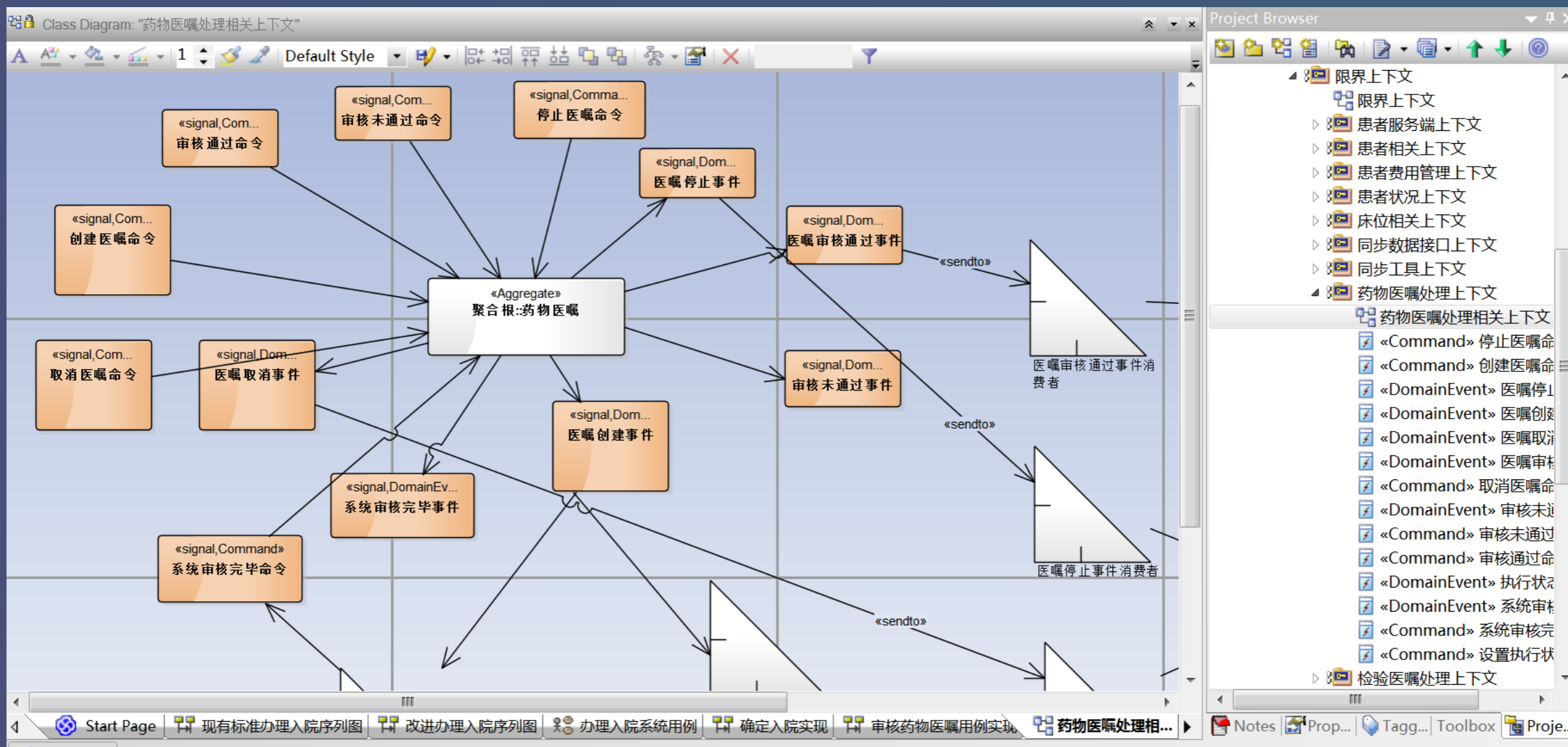
分析模型



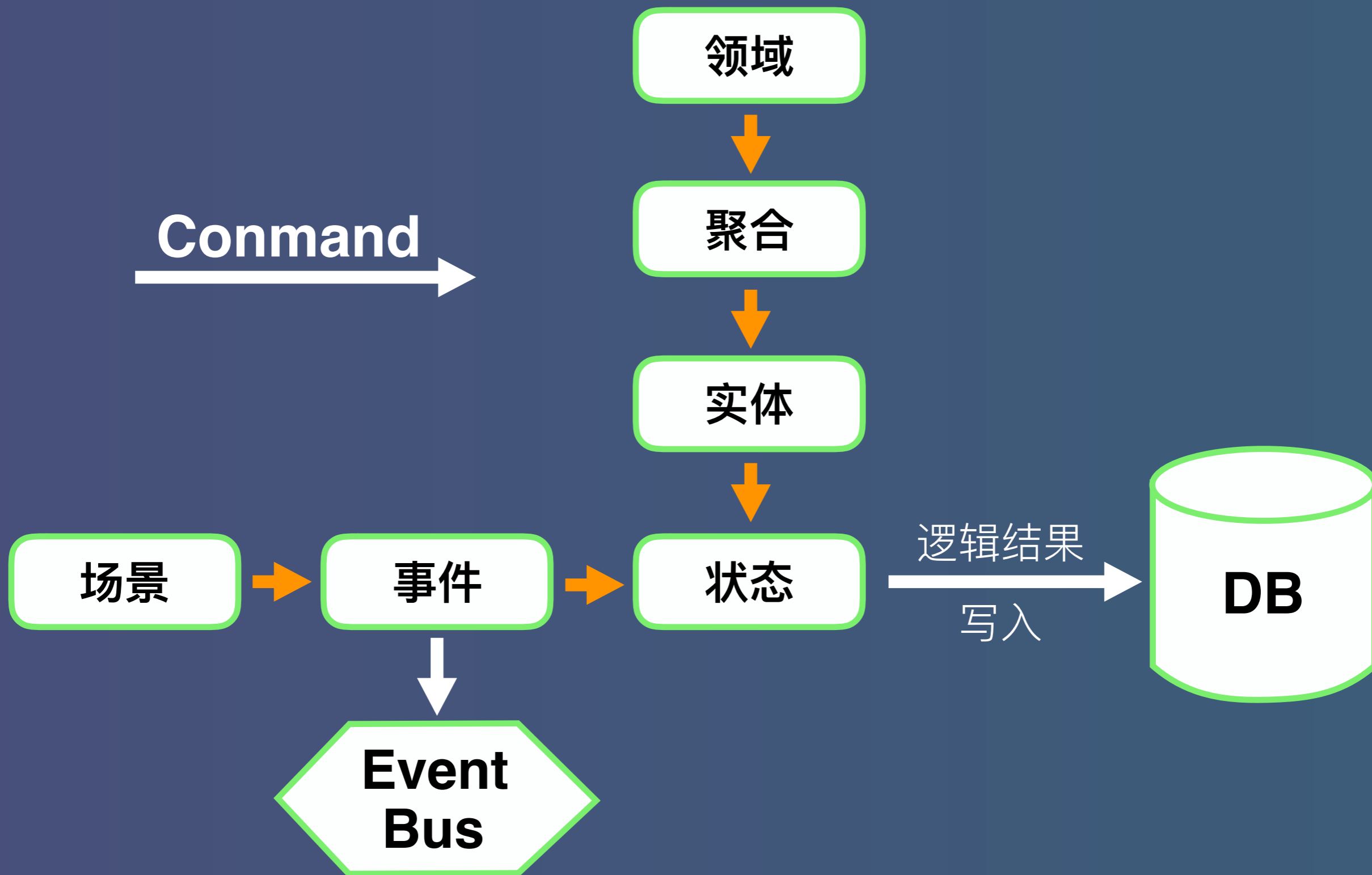
分析模型



分析模型



分析方法



架构

EDA

CQRS

Microservice

EDA

事件代表过去发生的事件，事件既是技术架构概念，也是业务概念，以事件为驱动的编程模型称为事件驱动架构 EDA。

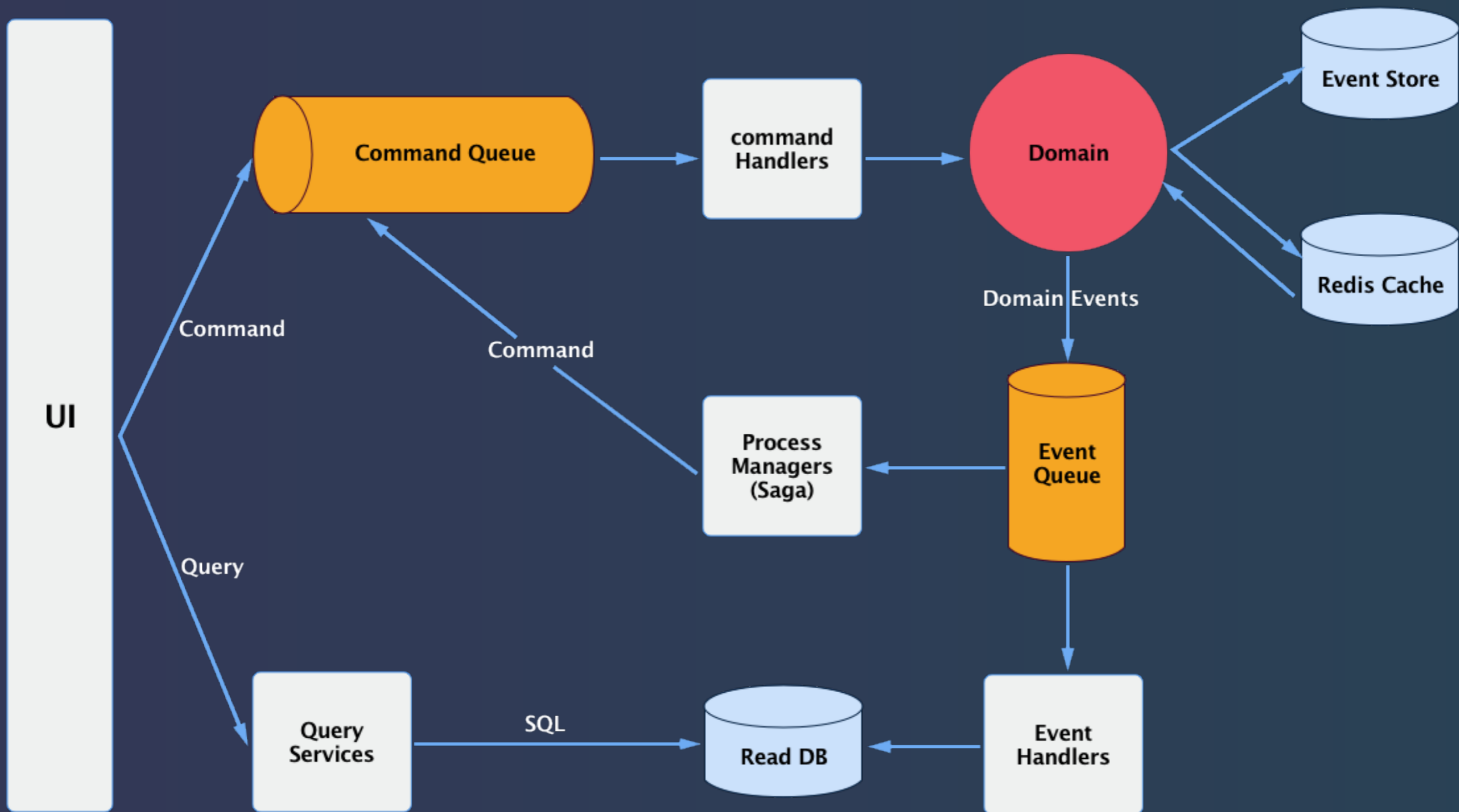
EDA 架构的三个特性：

异步

实时

彻底解耦

CQRS



Domain Event

在Evans DDD实现过程中，经常会碰到实体和服务Service以及Repository交互过程，这个交互过程的实现是一个难点，也是容易造成贫血模型的主要途径。

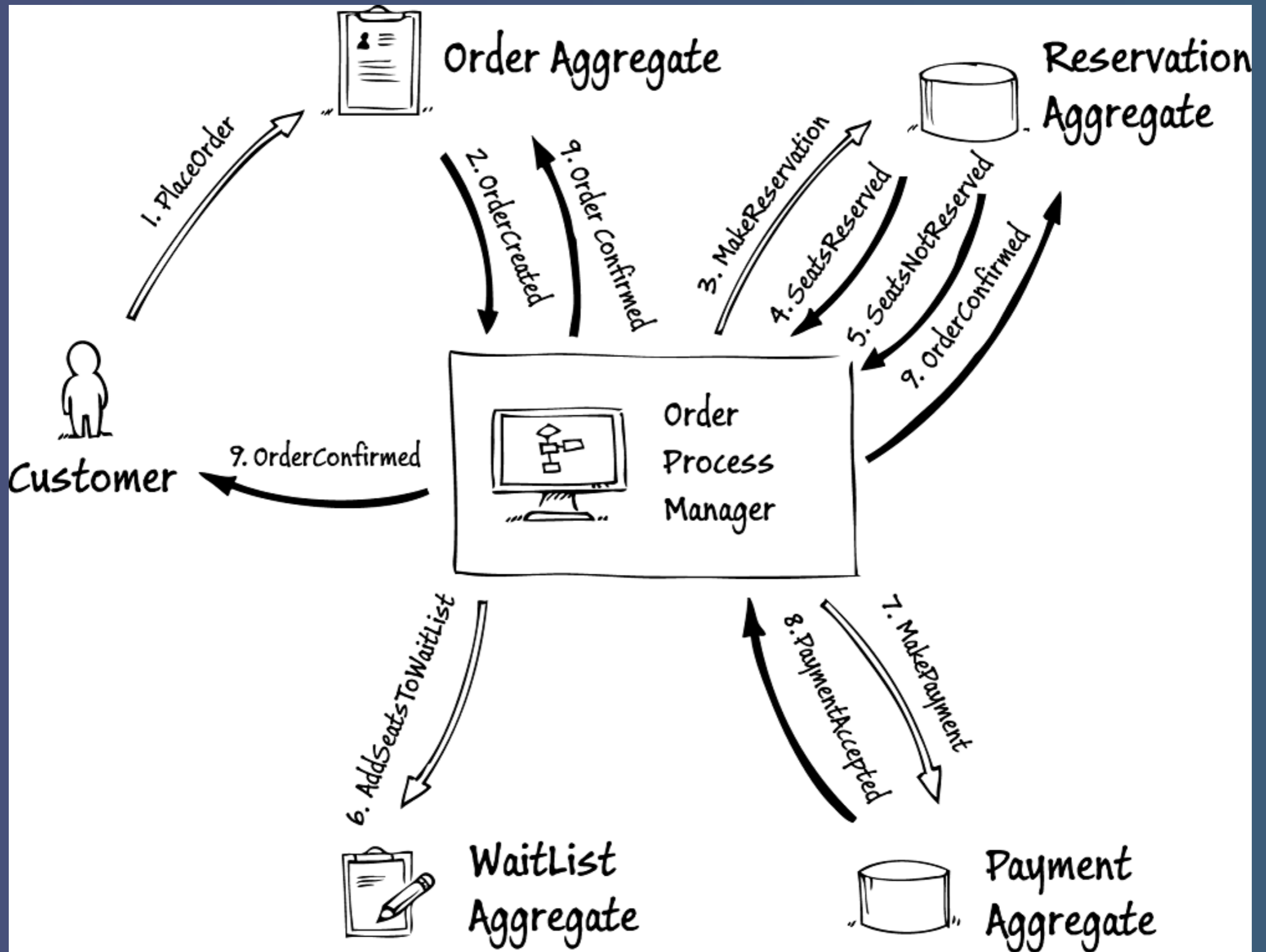
如果把这个业务方法放入服务，就容易造成实体的贫血；但但是如果把服务注射到实体中，也非常丑陋。

领域中处理一些业务逻辑后，就会对领域对象的状态做出一些改变，领域事件的订阅者去完成领域对象状态改变之后的操作。

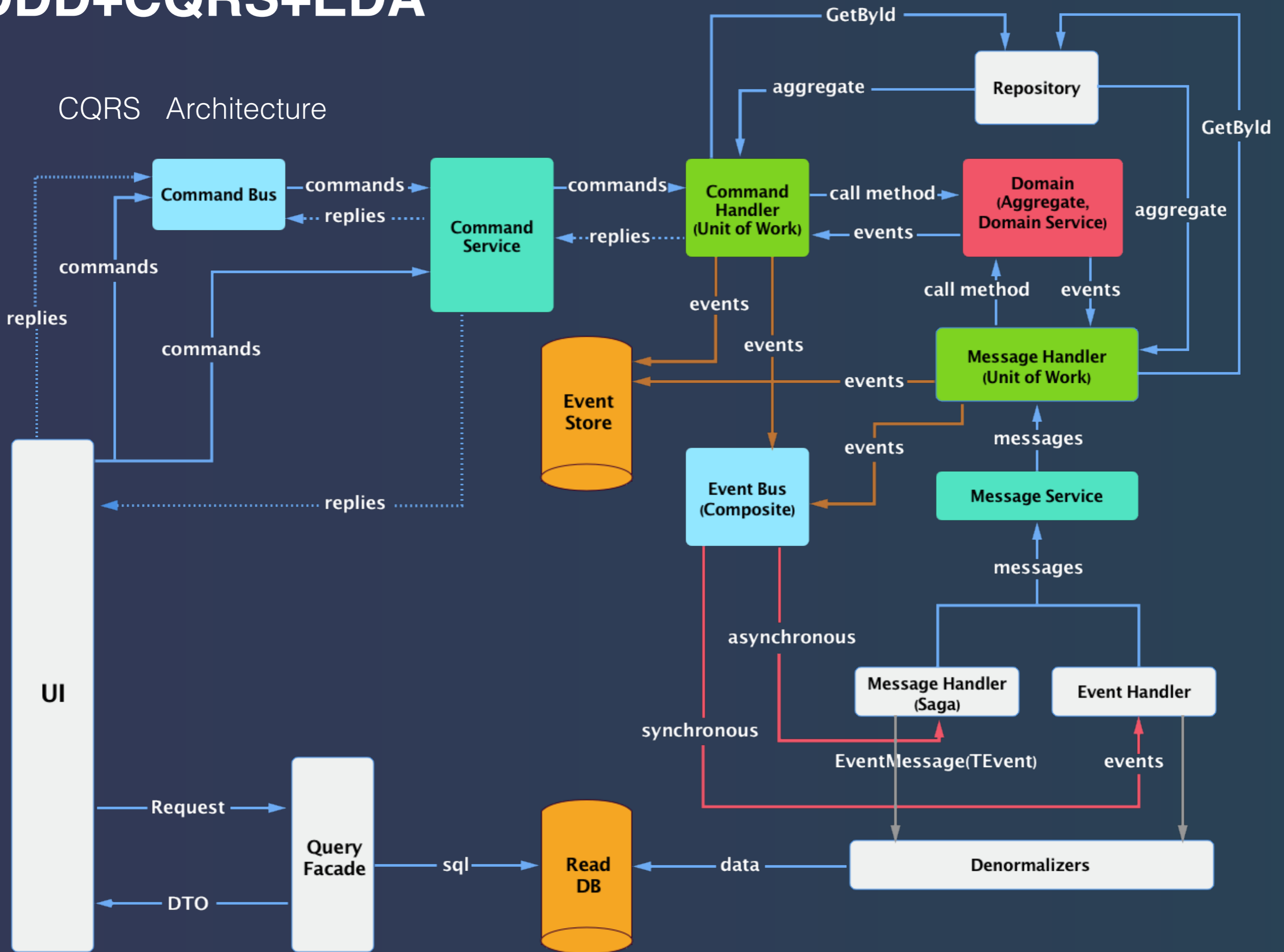
Saga

事件驱动的、分布式的并行处理模式。

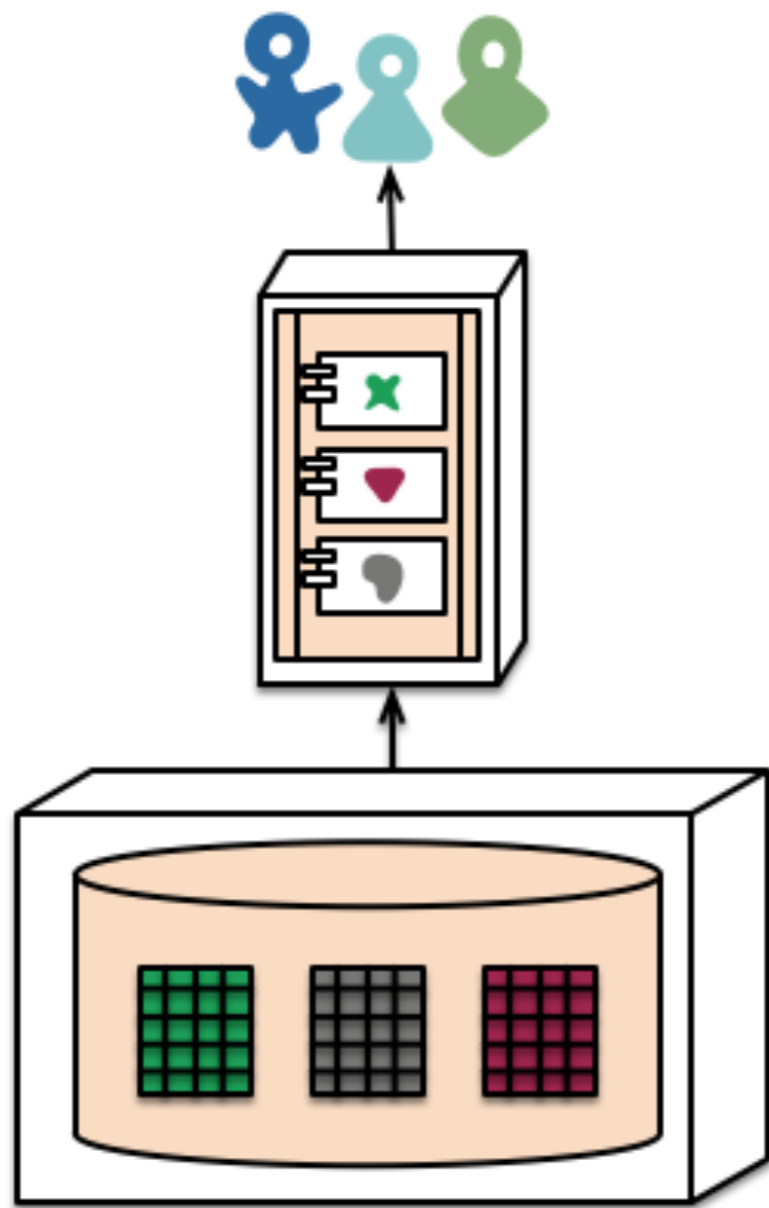
我们一般会用它处理多个聚合根交互的业务逻辑



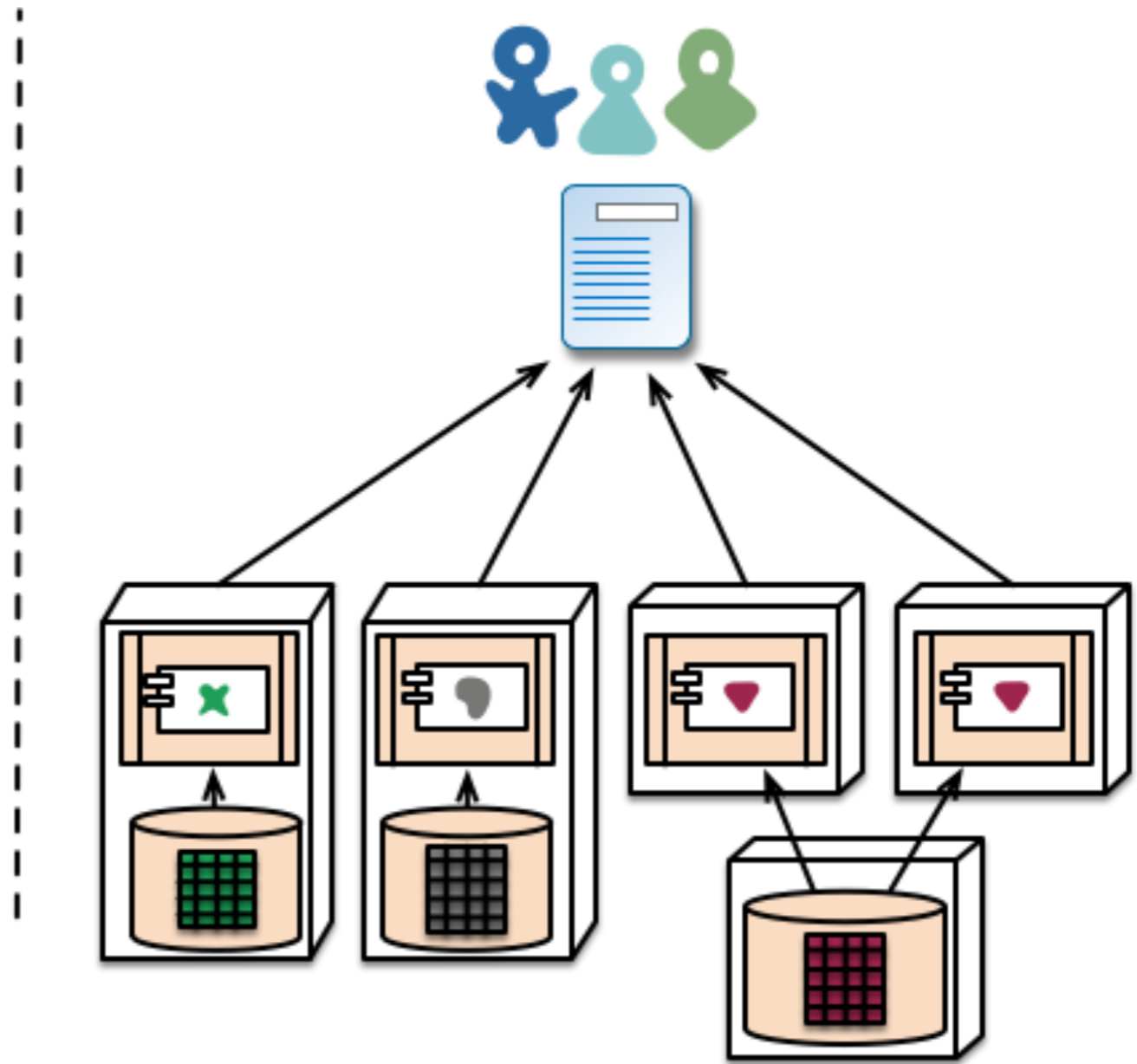
DDD+CQRS+EDA



Microservice



monolith - single database



microservices - application databases

Advantages of Monolithic Applications

使用广泛

Well Known

IDE支持丰富

IDE-friendly

共享方便

Easy Sharing

测试简单

Simplified Testing

部署简单

Easy Deployment

Disadvantages of Monolithic Applications

缺乏敏捷

Limited Agility

持续交付不方便

Obstacle for continuous delivery

技术锁定

“Stuck” with Technology Stack

技术债务

Technical Debt

Characteristics of Microservices

领域驱动设计

Domain Driven Design

单一职责原则

Single Responsibility Principle

明确的接口

Explicitly Published Interface

独立部署、升级、扩展、替换

Independently Deploy, Upgrade, Scale, Replace

技术多样性

Potentially Heterogeneous/Polyglot

轻量级通信

Light-weight Communication

Advantages of Microservices

容易开发、理解和维护

Easier to develop, understand, and maintain

快速启动 Starts faster than a monolith

方便部分升级

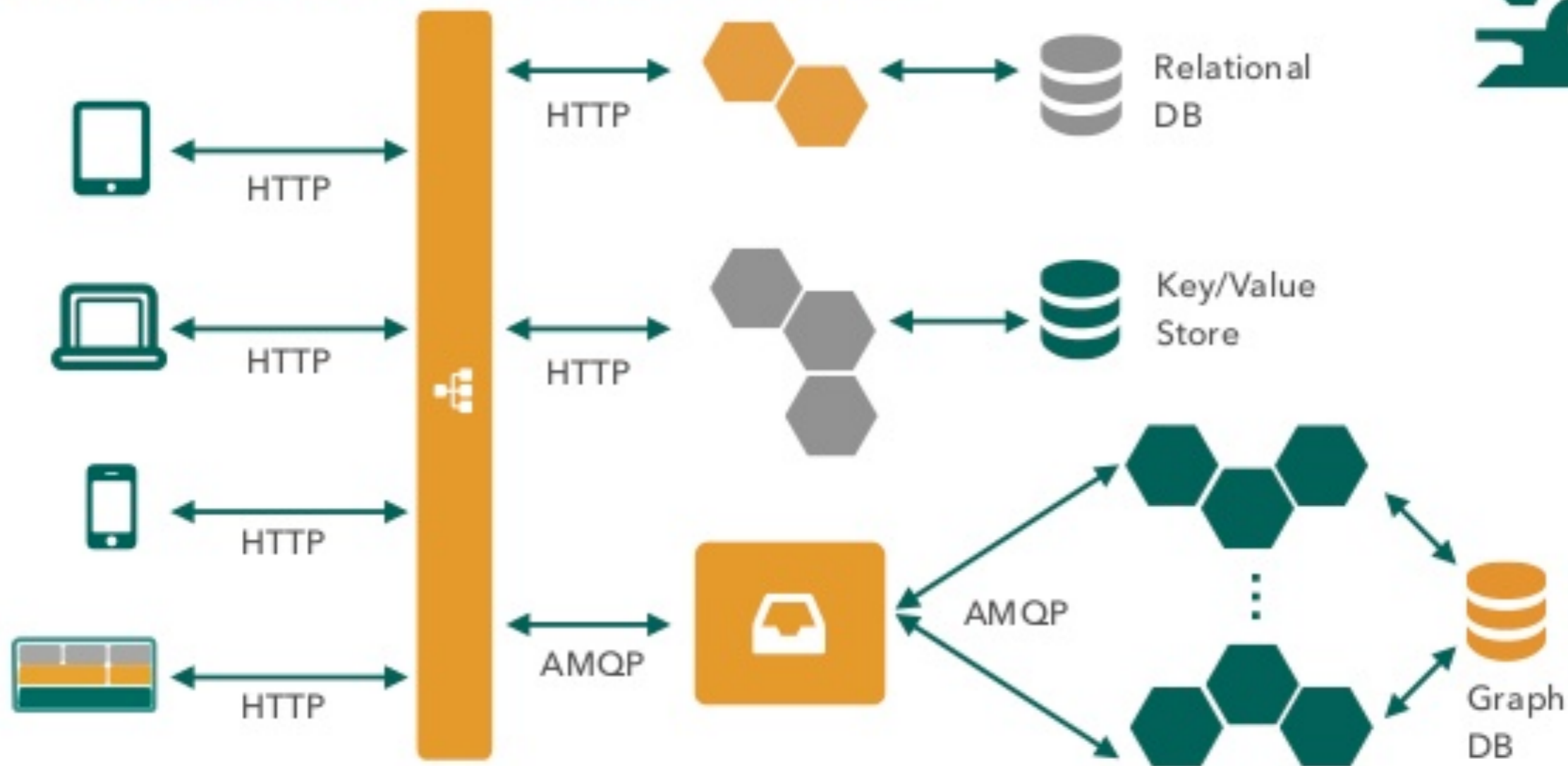
Local change can be easily deployed

独立扩展 Scale independently

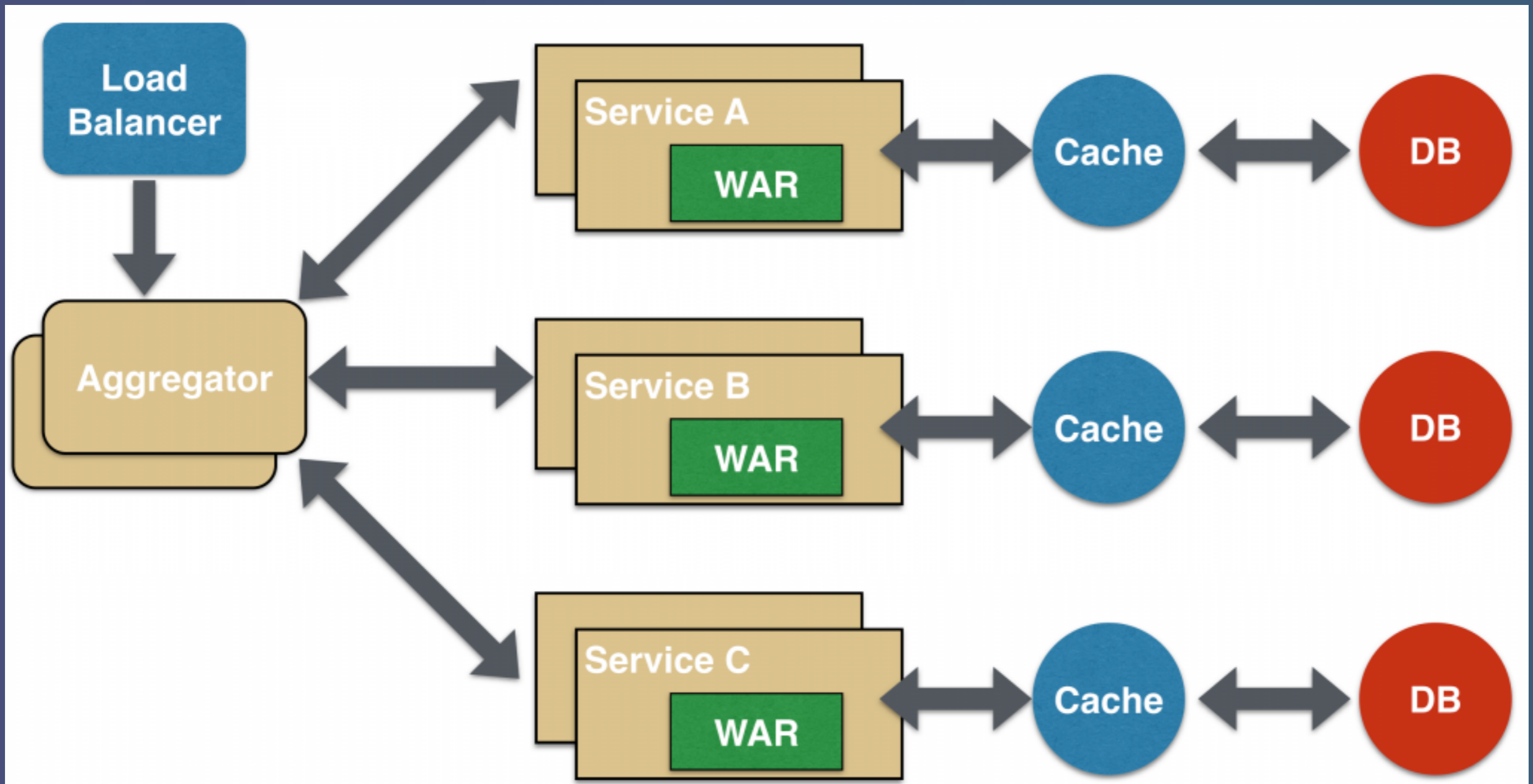
故障隔离 Improves fault isolation

架构选择更自由 No long term commitment to any stack

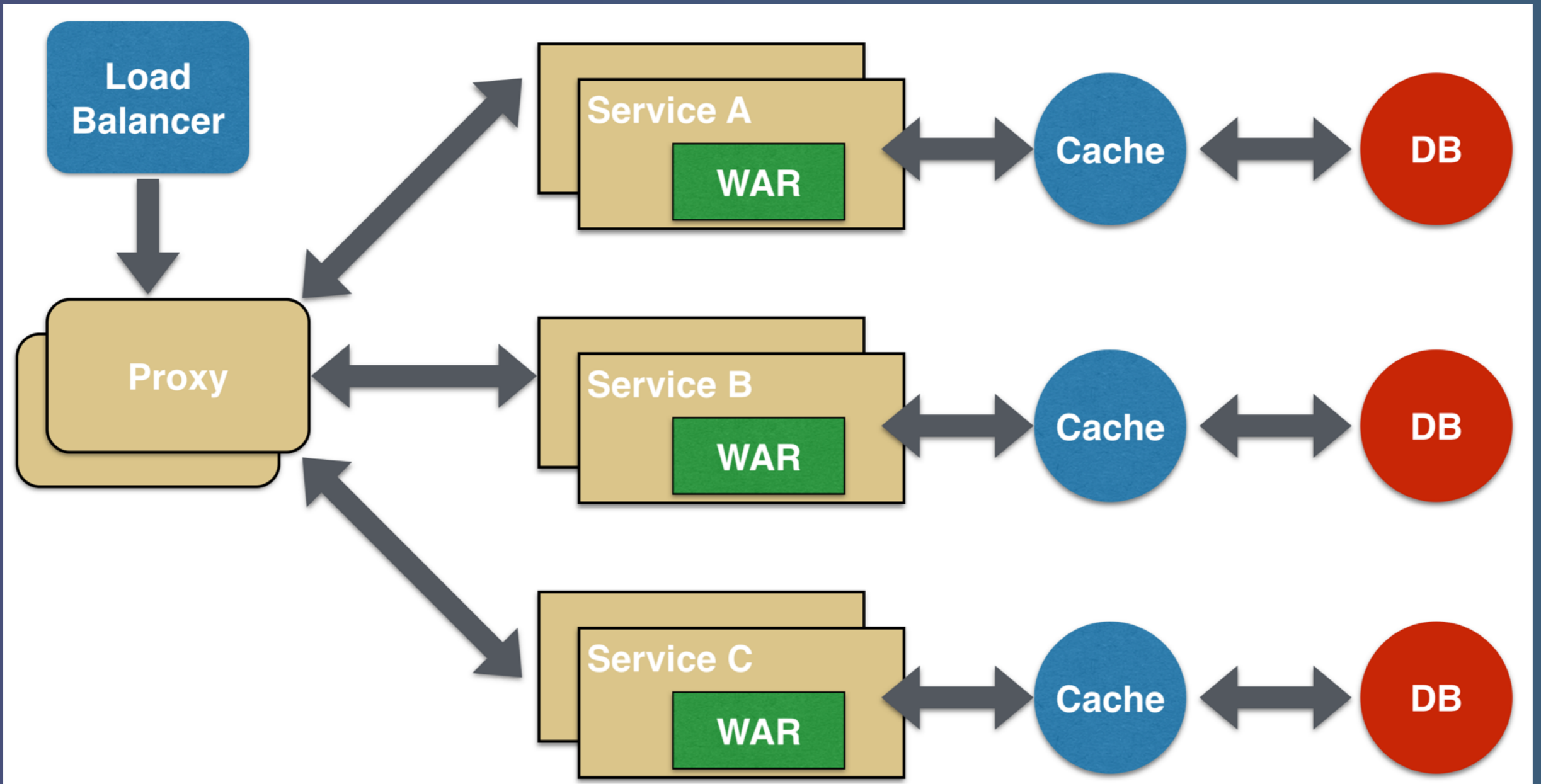
Microservice Architecture



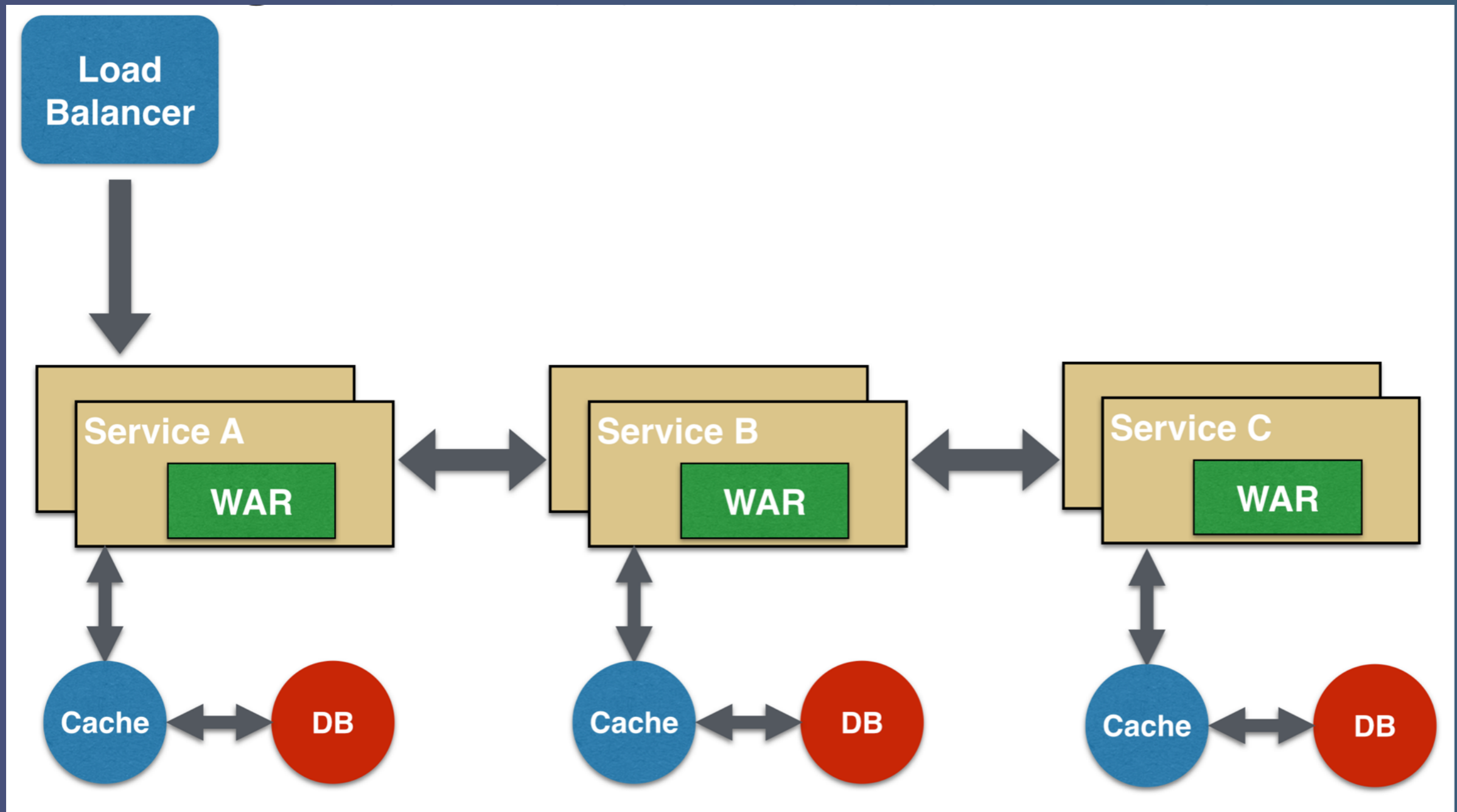
Aggregator Microservice Design Pattern



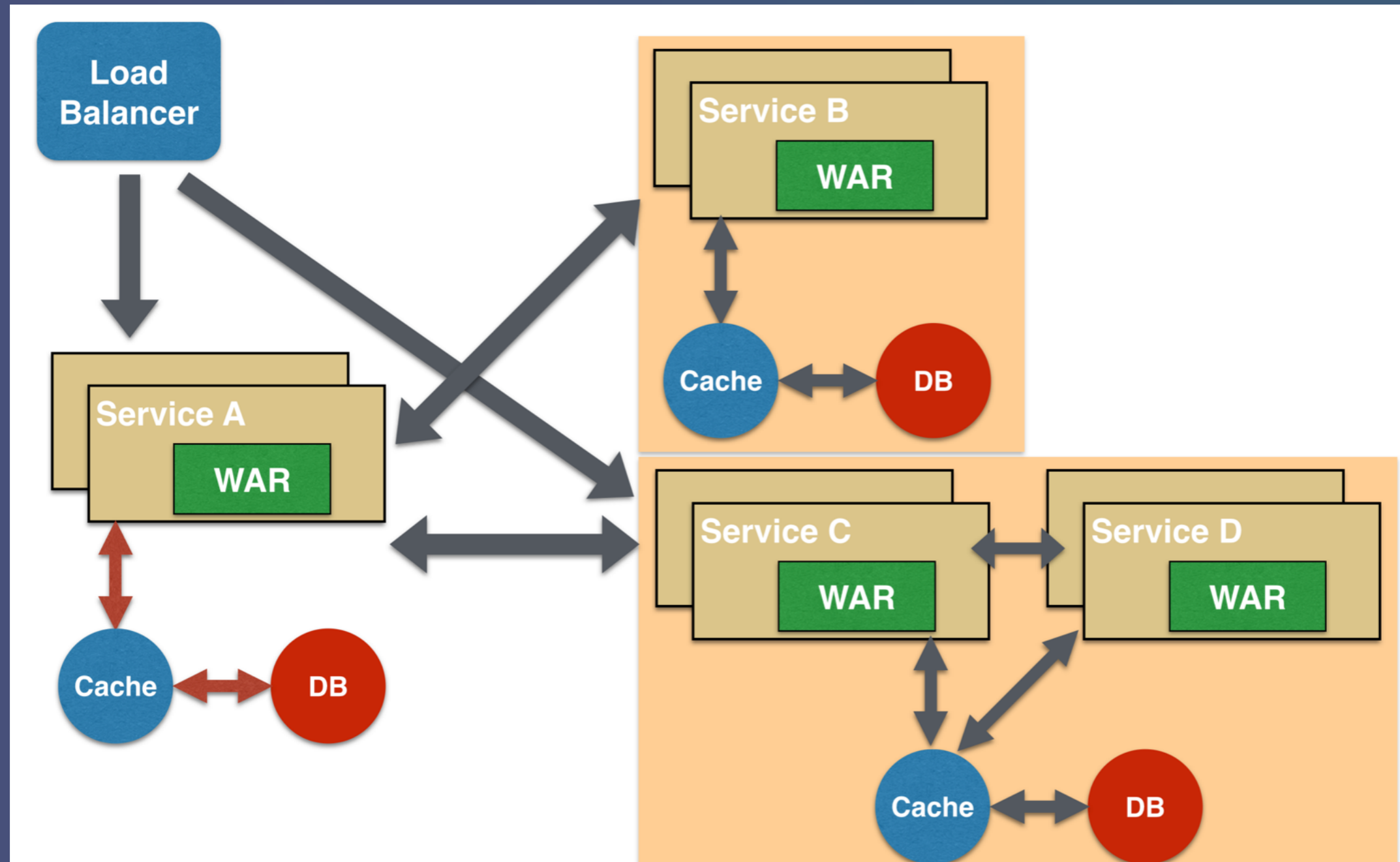
Proxy Microservice Design Pattern



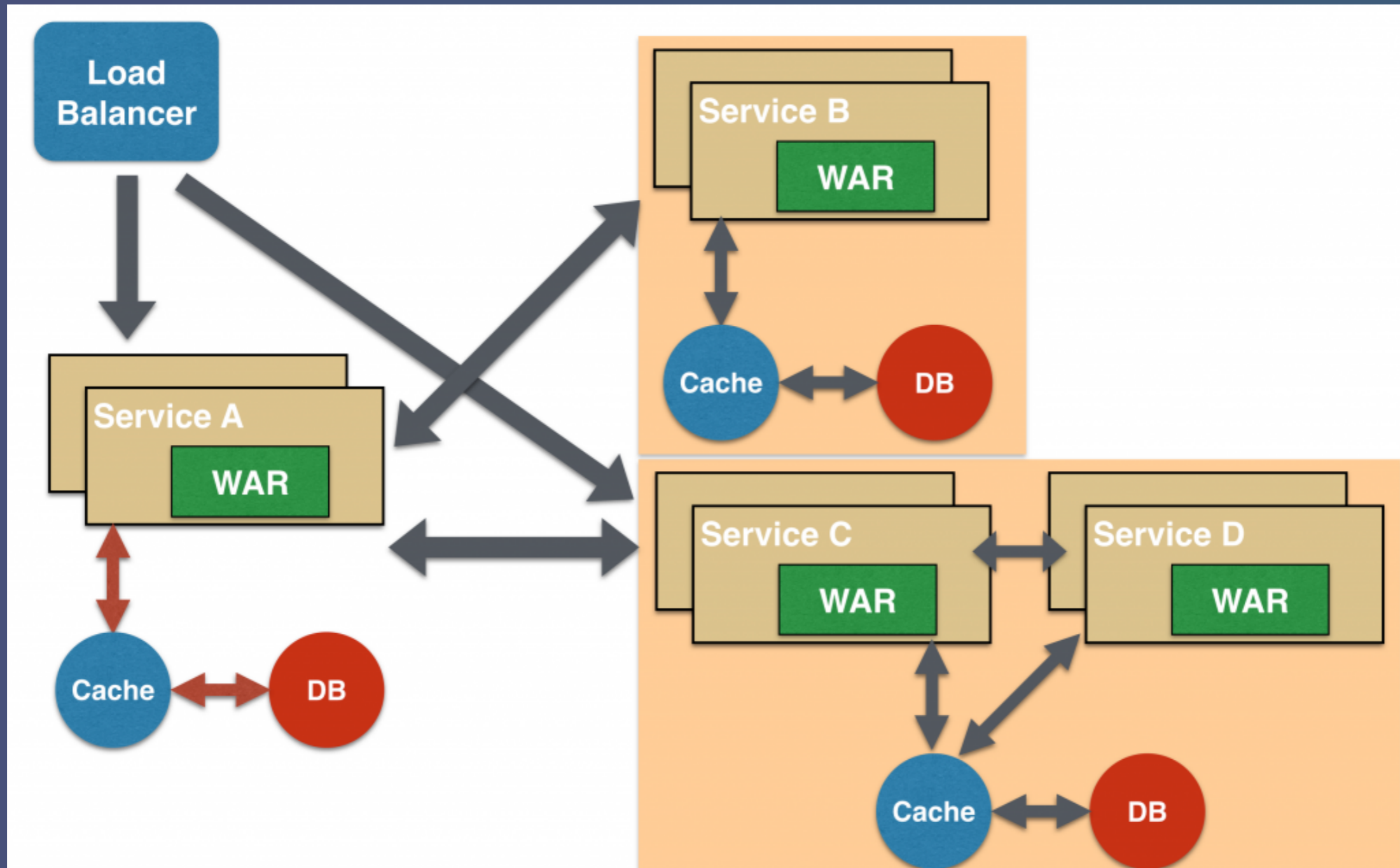
Chained Microservice Design Pattern



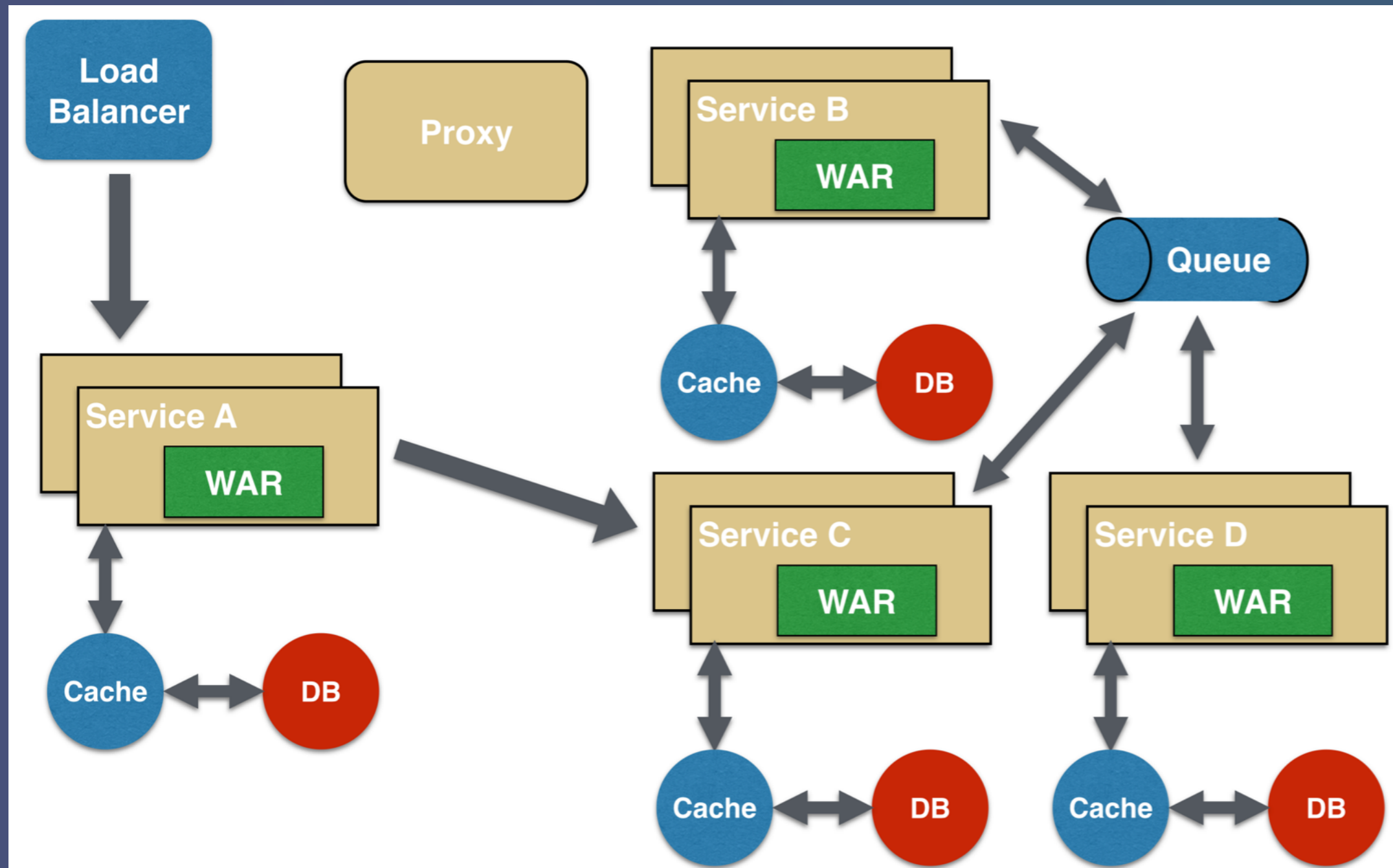
Branch Microservice Design Pattern



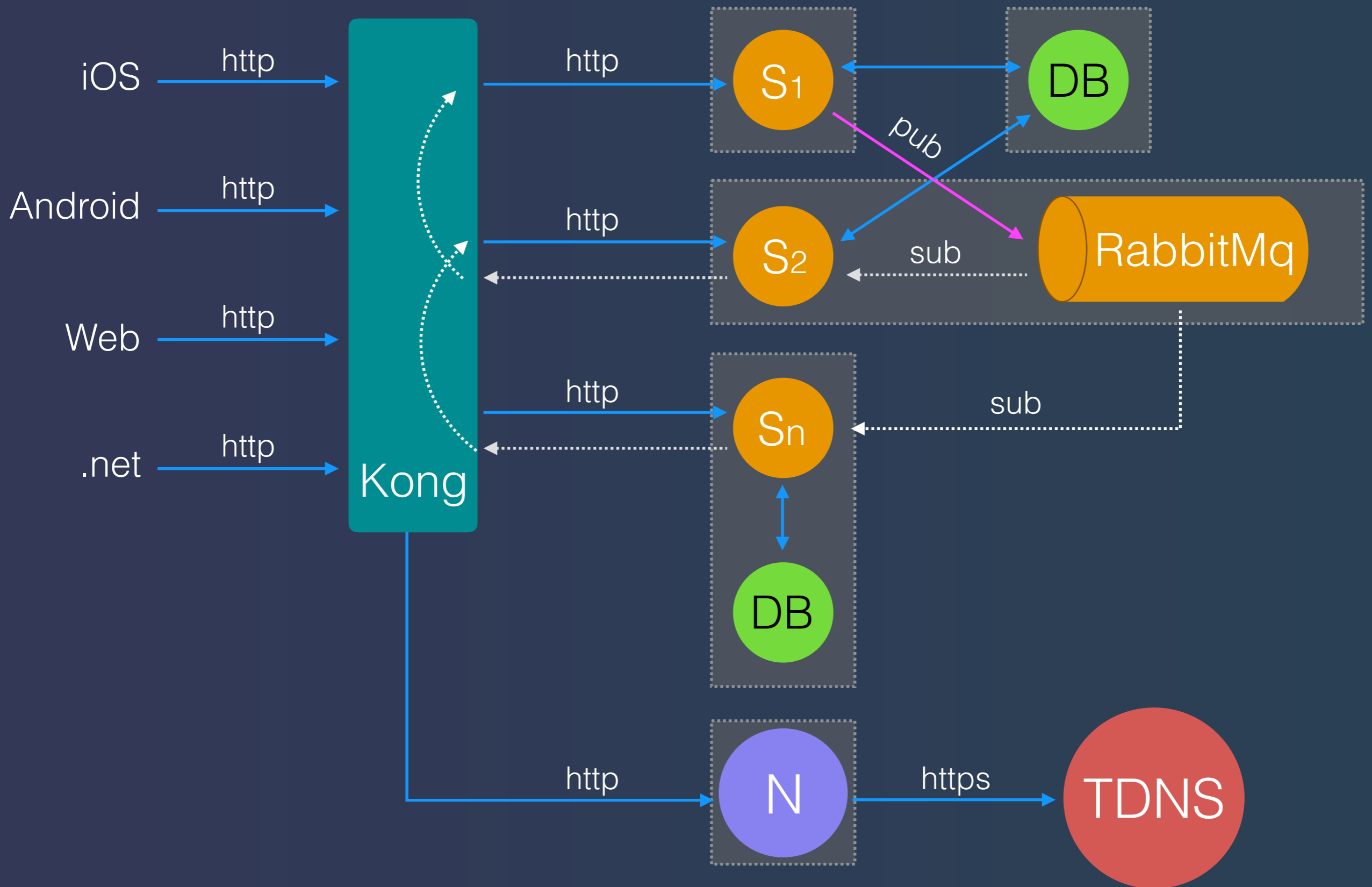
Shared Data Microservice Design Pattern



Asynchronous Messaging Microservice Design Pattern



我们的架构



参考

<http://martinfowler.com/articles/microservices.html>

<http://blog.arungupta.me/microservices-monoliths-noops/>

<http://blog.arungupta.me/microservice-design-patterns/>

Docker



No Docker No Microservices

部署代码化

InfrastructureAsCode

CI/CD

Dockerfile

Docker Compose

Caas平台选择



DaoCloud



DaoCloud

最期望的功能

支持Docker Swarm

支持Docker Compose 2.0

谢谢大家!

招贤纳士

全职/实习生

学习能力强、好奇心强！

